P RTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

stack operands "garbage collection" SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **stack operands garbage collection** Found **4,111** of **192,172**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Adaptive techniques: Optimistic stack allocation for java-like languages
Erik Corry
June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**
Publisher: ACM Press
Full text available: pdf(155.23 KB) Additional Information: full citation, abstract, references, index terms

Stack allocation of objects offers more efficient use of cache memories on modern computers, but finding objects that can be safely stack allocated is difficult, as interprocedural escape analysis is imprecise in the presence of virtual method dispatch and dynamic class loading. We present a new technique for doing optimistic stack allocation of objects. Our technique does not require interprocedural analysis and is effective in the presence of dynamic class loading, reflection and exception han ...

Keywords: Java, garbage collection, stack allocation

2 Fast, effective code generation in a just-in-time Java compiler
Ali-Reza Adl-Tabatabai, Michał Cierniak, Guei-Yuan Lueh, Vishesh M. Parikh, James M. Stichnoth
May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation PLDI '98**, Volume 33 Issue 5
Publisher: ACM Press
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

A "Just-In-Time" (JIT) Java compiler produces native code from Java byte code instructions during program execution. As such, compilation speed is more important in a Java JIT compiler than in a traditional compiler, requiring optimization algorithms to be lightweight and effective. We present the structure of a Java JIT compiler for the Intel Architecture, describe the lightweight implementation of JIT compiler optimizations (e.g., common subexpression elimination, register allocation, and elim ...

3 Techniques for obtaining high performance in Java programs
Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3
Publisher: ACM Press

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	50	(US-20020107667-$ or US-20020073283-$ or US-20010013117-$ or US-20040073897-$ or US-20040167947-$ or US-20040187102-$ or US-20050044540-$ or US-20060050885-$ or US-20030028742-$ or US-20040153827-$ or US-20050138649-$ or US-20040193828-$ or US-20030221047-$).did. or (US-6081665-$ or US-6026237-$ or US-6151703-$ or US-6247020-$ or US-6253215-$ or US-6442663-$ or US-6594820-$ or US-6829686-$ or US-6282702-$ or US-6093216-$ or US-6138127-$ or US-6718438-$ or US-6327701-$ or US-5925123-$ or US-6047125-$ or US-5241673-$ or US-6192517-$ or US-6308315-$ or US-6415302-$ or US-6424977-$ or US-6434576-$ or US-6434577-$ or US-6442751-$ or US-6449626-$ or US-6735761-$ or US-7069281-$).did. or (US-7092978-$ or US-6651186-$ or US-6883163-$ or US-6981245-$ or US-6986132-$ or US-6308317-$ or US-5440746-$ or US-6473777-$ or US-6735680-$ or US-6317872-$ or US-6807551-$).did.	US-PGPUB; USPAT	OR	ON	2006/11/03 16:36
S2	233	712/202	US-PGPUB; USPAT	OR	ON	2006/11/03 16:36
S3	1441	((707/206) or (712/202) or (717/146-148)).CCLS.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/11/03 16:40
S4	17	stack same operand same method same spill	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/03 16:41
S5	1	("6058457").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2006/11/03 16:41

EAST Search History

S6	3	((java adj card) javacard) with garbage adj collect$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 13:36
S7	1	(evaluation adj stack) with garbage adj collect$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:15
S8	5	(evaluation adj stack) same garbage adj collect$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:20
S9	0	(evaluation adj stack) same operand same bytecode	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:21
S10	32	(evaluation adj stack) same operand	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:28
S11	6	stackmap	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:51
S12	20	(stackmap typemap)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:51
S13	3	(stackmap typemap) with method	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 14:52

EAST Search History

S14	1	(stackmap typemap) same method with block	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 16:12
S15	0	"gosling property"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/14 16:12